# Multi-domain Virtual Content-Aware Networks Mapping on Network Resources

Eugen Borcoci, Radu Dinel Miruta, Serban Georgica Obreja
Telecommunication Dept.
University POLITEHNICA of Bucharest
Bucharest, Romania
eugen.borcoci@elcom.pub.ro

*Abstract* — For Future Internet multimedia distribution with different levels of guarantees a transport solution may consist in creation of parallel multi-domain Virtual Content Aware Networks (VCAN) constructed as overlays over IP networks. How to map the VCANs topologies and requested capacities on real core network domain resources both inter and intra-domain while optimizing the resource usage is still an open research issue. This paper continues previous studies for a hierarchical method to perform VCAN mapping while meeting QoS constraints for multi-domain VCANs.

***Keywords — Resource allocation, Content-Aware Networking, Network Aware Applications, Multi-domain, Inter-domain peering, Management, Future Internet.***

## I. INTRODUCTION

The Future Internet will have a strong content orientation, including multimedia flows distribution, [3-6]. Customising the media transport can be done by creating virtualized *Content Aware Networks* (VCAN) on top of the IP level and coupling the network layer more strongly with the upper layers by having *Network Aware Applications* (NAA). The Content Awareness (CA) novel concept means that new intelligent routers will process and forward the data, based on *content type* recognition or, even more, treating the data objects based on their *name* and not based on *location address*, [3] [4] [5]. The VCANs can be constructed based on virtualisation techniques, agreed to be used to overcome the ossification of the current Internet [1], [2].

The ALICANTE European FP7 ICT research project, “Media Ecosystem Deployment Through Ubiquitous Content-Aware Network Environments”, [6] [7], [8], adopted the NAA/CAN philosophy and defines an architecture, on top of multi-domain IP networks, to offer services for different business actors playing roles of consumers and/or providers. The ALICANTE defines several cooperating environments: *User (UE), Service (SE)* and *Network (NE).* The UE contains the End-Users (EU) terminals and SE contains High Level Service Providers (SP) and Content Providers (CP). The NE contains a new CAN Provider (CANP) to manage and offer VCANs and traditional Network Providers (NP/ISP) - managing the network at IP level. On demand of SP, the CANP creates VCANs unicast and multicast (QoS enabled) over multi-domain, multi-provider IP networks. The network resources are provided by the NPs. They are managed quasi-statically by provisioning and also dynamically by using adaptation procedures for media flows. The management is based on Service Level Agreements (SLAs) negotiated and concluded between providers (e.g. SP, CANP). In the Data Plane, content/service description information (metadata) can also be inserted in the media flow packets by the Content Servers and treated appropriately by the intelligent routers of the VCAN.

This paper studies the problem of mapping the overlay VCANs (as requested by SP) onto real network resources in a multi-domain context, while satisfying QoS constraints. It continues the work presented in [19]. The VCAN resources are first logically reserved; later when installation is requested by the SP, they will be really allocated in routers. Section 2 presents samples of related work. Section 3 summarizes the overall ALICANTE architecture VCAN general Management and inter-domain peering solution. Section 4, presents a continuation of [19] and refines the algorithm of mapping the overlay VCANS topology onto network resources with constrained routing and resource reservation while optimizing the network resource usage. Section 5 contains some conclusions and future work outline.

## 2. RELATED WORK

The virtual networks optimal mapping on the resources offered by a physical IP Network is an NP-hard problem, [18]. It is of current interest given the increasing interest in virtualization. Our goal is to develop algorithms to map the QoS capable VCANs over several independent network domains, in a scalable way to finally assure efficient transport of real-time and media traffic. The solution should fit the ALICANTE system architecture where CAN Managers and Intra-domain Network Resources Managers exist associated to network domains– each of them supposed to be aware on the status of their resources. There are two solutions: find some inter-domain paths with/without checking the QoS properties. After paths finding, a negotiation protocol should be run, [6], [7], [8], [13], [14], between domain managers to establish inter-domains SLAs. If no QoS constraints are used during routing there are significant chances that the SLA negotiation will fail. So, it is a better solution to search for QoS enabled paths, [9], [10], [11], [12].

In ALICANTE an inter-domain overlay QoS peering and routing [7], [19], has been defined. An inter-domain overlay network is first defined, abstracting each domain with a node and its inter-domain links. Protocols are needed to transport QoS and other information between nodes and based on this information, QoS routing algorithms can choose QoS capable paths.

For inter-domain signaling several solutions are proposed (cascade, hub, mixed-mode), [13], [14], [16]. However, they do not consider the content awareness capabilities of the multiple domain infrastructures, nor the virtualization aspects. The solution proposed here is hierarchical, applicable both to inter-domain and intra-domain context: QoS enabled (constrained) routing based on overlay topology, admission control check and then VCAN virtual links assignment to real paths.

## 3. ALICANTE SYSTEM ARCHITECTURE AND VCAN MANAGEMENT

The ALICANTE concepts and architecture are defined in [6][7][8]. Figure 1 shows a simplified picture, emphasizing the actors: the network contains several Core Network Domains (CND), belonging to NPs (can be Autonomous Systems - AS) and access networks (AN). The ANs are out of scope of VCANs. The CAN layer M&C is partially distributed: one *CAN Manager* (CANMgr) belonging to CANP exists for each IP domain, doing VCAN planning, provisioning, advertisement, offering, negotiation installation and exploitation. Each domain has an *Intra-domain Network Resource Manager* (Intra-NRM), as the ultimate authority configuring the network nodes. The EU terminals are connected to the network through Home Boxes (HB). The novel CAN routers are called *Media-Aware Network Elements (MANE)* to emphasize their additional capabilities: content and context – awareness. The CAN layer cooperates with HB and SE by offering them CAN services.

The VCAN Management has been already defined in [8]. Here only a short summary is recalled for sake of clarity. A functional block at SP performs all actions needed for VCAN support (planning, provisioning, negotiation with CANP, VCAN exploitation). The CAN Manager (*CANMgr@CANP)* performs, at the CAN layer, VCAN provisioning and operation. The two entities interact based on the SLA/SLS contract initiated by the SP. The main interactions in the Figure 1 are denoted with 1, 2, 3, 4 and explained in [19]. The content awareness (CA) is explained in [7], [8], and it is not the scope of this paper to be treated in details. It is realized by cooperation between Data plane and Management and control Plane.

The Data plane topology in multi-domain context can be of any kind. The *M&C topology* defines how the CANMgrs associated to CNDs inter-communicate for multi-domain VCANs construction. The VCAN initiating CANMgr has to negotiate with other CAN Managers. The *hub model* was selected in ALICANTE and justified, [19][20]Two functional components are needed: (1) inter-domain topology discovery protocol; (2) overlay negotiation protocol for SLA/SLS negotiations between CAN Managers.

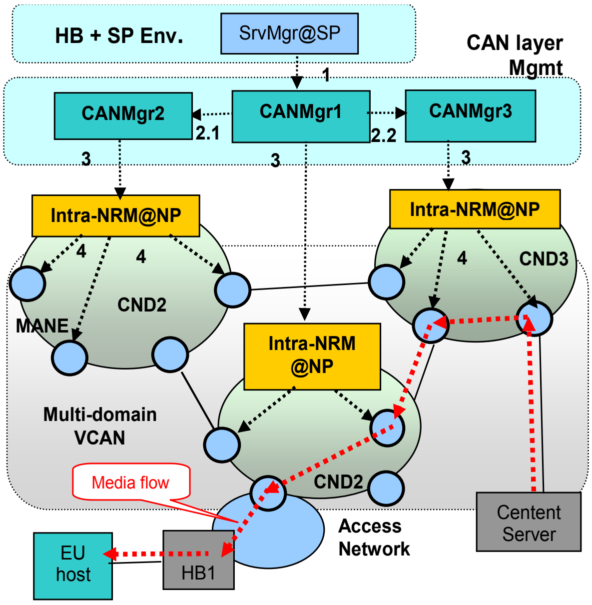

Figure 1 Example of a multi-domain VCAN. Data Plane topology and management interactions

An inter-domain discovery protocol [19], has informed each CANMgr about the inter-domain graph where each CND is abstracted as a node (the inter-domain links capacities are also learned). The SP asks for a VCAN to a CANMgr (Initiator) – see action 1. The SP knows the edge points of this VCAN, i.e., the MANEs IDs where different sets of HB currently are, or they will be connected. The initiator CANMgr determines all CNDs involved (from the SP information and its inter-domain knowledge) and then negotiate with all other CAN Managers (actions 2.1, 2.2) to agree and reserve resources for the VCAN. The split of the SLS parameters (if it is the case) should be done at the initiator (e.g. for delay). In a successful scenario, the multi-domain VCAN is agreed and then it will be later instantiated in the network.

The pair (CANMgr - Intra-NRM) knows about its network resources: abstract view of its CND and output links towards neighbors in a form of a set of virtual links (called *Traffic Trunks*). A set of such links can belong to a given QoS class. A multiple domain VCANs should also belong to some QoS class and therefore inter-domain QoS aware routing information is necessary in order to increase the chances of successful SLS establishment, between CANMgrs. The multi-domain VCANs deployment needs knowledge on a virtual multi-domain topology, [19]. Therefore in the following sections we suppose that this information is known by each CANMgr.

## 4. VCAN MAPPING AND RESOURCE RESERVATION

### 4.1 General Functions

The process of signaling between ALICANTE managers in order to construct VCANs is explained in [7], 8], [19]. Her we only use the results, in order to continue the algorithm development. The VCAN initiator Manager knows the inter-domain overlay network topologies (ONT) and inter-link capacities and it negotiates with SP an SLS contract and maps VCANs onto network resources. The VCAN mapping is done on two hierarchical levels: inter-domain and intra-domain. For the first we suppose that based on ONT knowledge the initiator CAN Manager has determined which domain have to participate at VCAN and knows the inter-domain graph where each domain is abstracted as a node. Also inter-domain link capacities assigned to this VCAN are known. Note that in a more complex approach several QoS parameters can be considered, e.g. delay, but here we treat the simplified case of only capacity (bandwidth).

The problem is the following: given an inter-domain graph and a Traffic Matrix (requests contained in the SLS), how to map the TM onto real network graph as to respect the minimum bandwidth constraints and also optimize the resource usage. A similar problem is then solved for each intra-domain case using a similar approach. The initiator CAN Manager first splits the initial SLS among core network domains producing the set of SLS parameters valid to be requested to each individual CND. The *inputs* are: ONT graph, abstracting each CND by a node; QoS characteristics of the inter-domain links (bandwidth -only,; the delay-it is not yet considered in this first algorithm variant); Traffic Matrix (and other QoS information) of the SLS proposed by SP. The *outputs* are the Traffic matrices for each CND composing the VCAN. To do this, the initiator CANMgr will run a constrained routing algorithm. A combined metric will be proposed here for a link, similar to [9], considering the bandwidth request, the bandwidth available, targeting to choose the widest path.

The cost of a inter-domain link (i,j) in the ONT can be *C(i,j) = Breq/Bij= Breq/Bavail* , where Bij is the available bandwidth on this link and *Breq* is the bandwidth requested for that link. Another useful interpretation of this ratio is as link utilization factor; that is the alternative notations will be used: C(i,j) = $U_{link\_ij.}$ The metric should be $\leq 1$, this representing the bandwidth constraint applied to the algorithm. The metric is additive, so one can apply modified Dijkstra algorithm to compute the *Shortest Path Trees (SPT)* one tree for each ingress node where the traffic flows will enter. Note however that this metric *Breq/Bij* can be only computed if we know the mapping TT- link ( i.e we know *Breq* for a given link), which is not yet our case. The mapping is to be done jointly with the routing process. So in the first approximation we consider 1/Bij as an additive link metric.

### 4.2 Routing and reservation algorithm

The general operation of the algorithm is summarized as *(*the notation *//text* repesents comments):

```
1. Split the Traffic Matrix TM (requests) in several trees, 1/ingress node ( I1, I2, ...In)
2. On the current graph, repeat for 1 to n:
        2.1. Compute the DJ_SPT (root_I1) where DJ means Dijkstra algorithm;//Routing metric ~1/Bij
        2.2. Select the TM branches that can be satisfied (i.e Bij > Breq for that direction);/ Mapping and AC
        2.3 Reserve capacities for these branches (subtraction);// reduced graph
        2.4. Compute the overall utilization for each path reserved: Upath= Sum_links (Breq/Bavail) * NHF(path);
        2.4 List the unsatisfied branches;
3. Aggregate for all inputs, (satisfied and not satisfied branches) and compute VCAN utilization (sum over all paths mapped onto the real graph);
Optimisation: change order {I1, ..In} and repeat 1..3.
```

Note: NHF is a factor taking into account the number of nodes traversed.

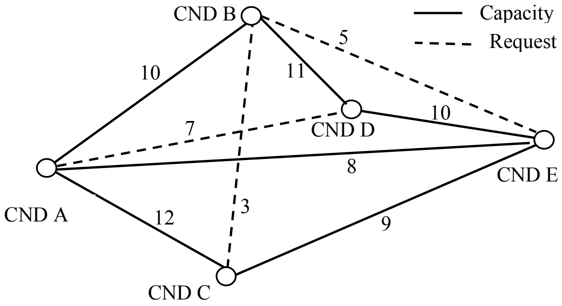

Figure 2 Network topology graph with a set of requests

To illustrate the combined routing and reservation algorithm we consider an example. In Figure 2 we have a non-oriented graph, where each node represents a Core network Domains (CND) and the inter-domain links are labeled with capacities. The dotted lines represents SP requests, expressed as a traffic matrix (TM: A-D 7MBps, B-E 5MBps and B-C 3Mbps). The objective is to select paths based on the criterion of largest capacity (bandwidth) possible and then make resource reservation on those paths.

The input data are: the topology matrix *mat(i,j)* Figure 3a, and the request matrix *cer(i,j),* Figure 3b. Element [i,j] of the mat(i,j) shows the link capacities allocated by bthe network managers for this VCAN. The request matrix shows on each row the source of request, the destination and the capacity requested.

$$\mathrm{mat(i,j)} = \begin{pmatrix} 0 & 10 & 12 & 0 & 8 \\ 10 & 0 & 0 & 11 & 0 \\ 12 & 0 & 0 & 0 & 9 \\ 0 & 11 & 0 & 0 & 10 \\ 8 & 0 & 9 & 10 & 0 \end{pmatrix} \qquad \mathrm{cer(i,j)} = \begin{pmatrix} 0 & 3 & 7 \\ 1 & 2 & 3 \\ 1 & 4 & 5 \end{pmatrix}$$

Figure 3 Network matrix (a) Requested matrix (b)

The specified algorithm has been implemented in C, using Visual Studio C++ Express Edition as development environment. The hardware platform used, relevant for the obtained results (especially the average processing time), is a device equipped with Intel(R) Core(TM)2 CPU T5600@1.83GHz processor and 2,00 GB installed memory (RAM) on a 32-bit OS.

The algorithm starts with splitting the set of requests in subsets characterized by a common source node, i.e. group 1 : 0-3, request 7, group 2: 1-2, request 3, 1-4, request 5 (the nodes A,B,C, .. are denoted as 0, 1, 2, ..). If SP does not specify a wanted order/priorities for request, the algorithm will process them in the received order. An SPT is computed (Dijkstra, using the metric 1/B) for each source node. Then meeting a request means finding the paths on SPT satisfying the requested bandwidth. The graph is updated after each solving of a request by subtracting the reserved bandwidth. SP will be finally informed about the requests non-satisfied and this is a matter of further negotiation SP-CANP. The steps are repeated for all other subsets as described in the summary of the algorithm.

### 4.3 Complexity and Optimisations

It is known that Dijkstra's original algorithm runs in $O(|V|^2)$ complexity; if the implementation is based on a min-priority queue implemented by a Fibonacci heap and then one has $O(|E| + |V| \log |V|)$ (Fredman & Tarjan 1984). The Dijkstra is the fastest (asymptotically) SPT algorithm for arbitrary directed graphs with nonnegative weights, [21]. Therefore the algorithm proposed in this work will have n* O(Dijkstra) complexity. However, the Network Provider would like the best VCAN mapping and to get the least overall utilization. A simple optimisation method is to recompute the step 2 of the algorithm for other order of inputs given by the bijective function f( I1, ..In) -> {Ik1, Ik2, ..Ikn} which creates actually permutations of the set {I1, ..In}. The function is random. The final mapping solution will be the one having the least overall utilization. The overall complexity will be n*n!* O(Dijkstra) which has not so good scalability.

We present some pragmatic solutions to improve the performance: 1. to stop repetitions of the step 2 if the overall utilization fulfill some enough good thresholds fixed by local CANP policy; 2. if the SP assigns a priority order to its requests, then no permutations are needed; 3. if no priority is given by SP, then one can define a local order to process the requests in increasing order of their average bandwidth for each ingress node (here we consider the policy to serve first the least demanding requests which consume less resources); if the last ones are not satisfied then, maybe the SP could accept a partial fulfillment of its high bandwidth requests)

*However note that in ALICANTE context this algorithm does not have to run in real time given that it is used at provisioning actions.* Therefore if the pragmatic optimizations are applied, its complexity is not a critical issue.

### 4.4 Preliminary Evaluation Results

This section presents some relevant but still preliminary results. To analyse the performance several input files have been generated containing different size of graphs with different loads using Graph Magics [20], different size of requests with different requirements. From Chart 1 we can observe that even if it is the same topology, the best cost varies from one processing order of requests to other one. In both of these orders the number of solved requests is the same (0.67 <-> 2/3, but at a different costs. More that that, even if the number of them is the same, the solved requests are different. In both cases the last request can not be honored, but this last request differs from one case to another.

As we expected, for the same network topologies, the time is increasing for a bigger number of permutations. In Chart 2 and Chart 3 the results correspond to the same network topologies but a different permutation number was set in the algorithm.

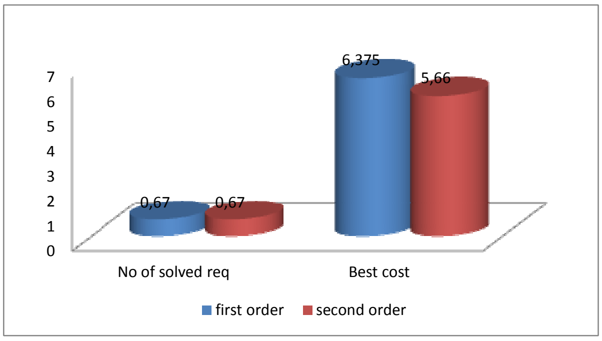

**Chart 1 – Different best cost value at different processing order of requests**

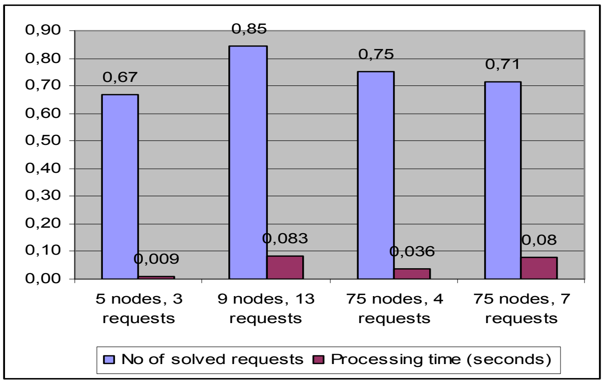

**Chart 2 – Time and number of solved requests vs. different topologies at the same number of permutations (4)**

Chart 2 illustrates number of solved requests and processing time for different topologies with 4 permutations. It can be observed in Chart 3 versus Chart 2 for the first topology (5 nodes and 3 requests) that the processing time is bigger for 4 requests (Chart 2) vs. only 2 requests (Chart 3).

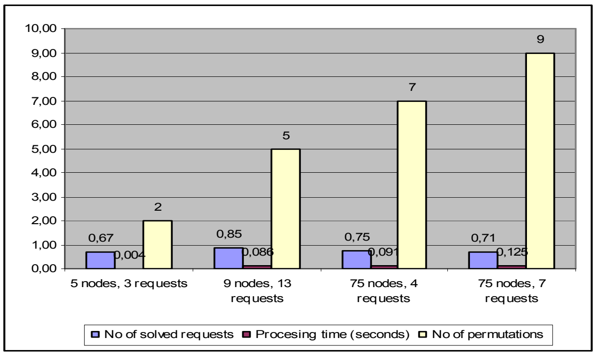

**Chart 3 – Time, number of solved requests vs. different topologies at different number of permutations**

We observed that for our generated topologies used in these tests, the number of solved requests is the same even on different number of permutations, but is not mandatory to be the same for all topologies.

## 5. CONCLUSIONS

This work proposed a combined algorithm to map the overlay VCANs onto multiple domain network resources, while respecting the QoS requirements issued by the Service Provider. An inter-domain part of the solution has been detailed in the paper by combining an overlay topology approach with an inter-domain constrained routing and admission control. Numerical examples obtained from preliminary implementations of the algorithm are given, showing the variability of performance with the graph complexity, number of requests and order of evaluation. Given the NP-hard characteristics of the problem, several efforts towards heuristic solutions improvement should still be done, while considering the SP and CANP policies. The solution is currently under complete design, evaluation and implementation inside the FP7 ALICANTE project. Further work will include more complex metrics and topologies and also comparative studies to other planning algorithms available in literature in more complex network topologies.

### Acknowledgments

This work was supported partially by the EC in the context of the ALICANTE project (FP7-ICT-248652) and partially by the projects POSDRU/89/1.5/S/62557 together with POSDRU/88/1.5/S/61178.

## REFERENCES

[1] T. Anderson, L. Peterson, S. Shenker, and J. Turner, “Overcoming the Internet Impasse through Virtualization”, Computer, vol. 38, no. 4, Apr. 2005, pp. 34–41.

[2] 4WARD, “A clean-slate approach for Future Internet”, http://www.4ward-project.eu/.

[3] J.Choi, J.Han, E.Cho, T. Kwon, and Y.Choi, A Survey on Content-Oriented Networking for Efficient Content Delivery, IEEE Communications Magazine, March 2011.

[4] V. Jacobson et al., “Networking Named Content,” CoNEXT ’09, New York, NY, 2009, pp. 1–12.

[5] W.K. Chai, et.al., CURLING: Content-Ubiquitous Resolution and Delivery Infrastructure for Next-Generation Services , IEEE Communications Magazine, March 2011.

[6] FP7 ICT project, “MediA Ecosystem Deployment Through Ubiquitous Content-Aware Network Environments”, ALICANTE, No248652, http://www.ict-alicante.eu/

[7] ALICANTE, Deliverable D2.1, ALICANTE Overall System and Components Definition and Specifications, http://www.ict-alicante.eu, Sept. 2011.

[8] E.Borcoci, M.Stanciu, D.Niculescu, D.Negru, G. Xilouris, “Connectivity Services Management in Multi-domain Content-Aware Networks for Multimedia Applications ” , Proc. of. INTERNET 2011, Luxembourg, June 2011.

[9] Zhi Li, P. Mohapatra, “QRON: QoS-Aware Routing in Overlay Networks”, IEEE Journal on Selected Areas in Communications, VOL. 22, NO. 1, January 2004, pp.29-39.

[10] Z. Wang, J. Crowcroft, “Quality-of-service routing for supporting multimedia applications”, IEEE Journal on Selected Areas in Communications, vol. 14, no. 7, pp. 1228—1234, 1996.

[11] J. Galán-Jiménez and A.Gazo-Cervero, “Overview and Challenges of Overlay Networks”, International Journal of Computer Science & Engineering Survey (IJCSES) Vol.2, No.1, Feb 2011, DOI : 10.5121/ijcses.2011.2102 19

[12] Z. Li, P. Mohapatra, and C. Chuah, Virtual Multi-Homing: On the Feasibility of Combining Overlay Routing with BGP Routing, University of California at DavisTechnical Report: CSE-2005-2, 2005.

[13] P. Levis, M. Boucadair, P. Morrand, and P. Trimitzios, “The Meta-QoS-Class Concept: a Step Towards Global QoS Interdomain Services”, Proc. of IEEE SoftCOM, Oct. 2004.

[14] M.P. Howarth, et. al., “Provisioning for Interdomain Quality of Service: the MESCAL Approach”, IEEE Communications Magazine, June 2005, pp. 129-137.

[15] L.FabioVerdi, and F. Magalhaes “Using Virtualization to Provide Interdomain QoS-enabled Routing”, Journal of Networks, April 2007, pp. 23-32.

[16] M. Boucadair, et al., “A Framework for End-to-End Service Differentiation: Network Planes and Parallel Internets”, IEEE Communications Magazine, Sept. 2007, pp. 134-143.

[17] Z. Wang, J. Crowcroft, “Quality-of-service routing for supporting multimedia applications”, IEEE Journal on Selected Areas in Communications, vol. 14, no. 7, pp. 1228—1234, 1996.

[18] A.Haider, Richard Potter and A. Nakao, “Challenges in Resource Allocation in Network Virtualization”, 20th ITC Specialist Seminar, 18.-20. May 2009, Hoi An, Vietnam, http://www.itcspecialistseminar.com/paper/itcss09Haider.pdf.

[19] E.Borcoci, R.Miruta, S.Obreja, “ Inter-domain Peering in Content-Aware Networks for Multimedia Applications” Proc. of. CTRQ 2012, France , April 2010

[20] http://www.graph-magics.com/

[21] http://en.wikipedia.org/wiki/Dijkstra%27s_algorithm